ESDIS Metrics System (EMS)

Page Tag Implementation Guide

Date: August 12, 2008

Version Number: 1.02

Verify that this is the correct version before use.

National Aeronautics and
Space Administration

This page is intentionally left blank.

Table of Contents

Table of Figures

This page is intentionally left blank.

1 Reference Materials

1.1 Document Version Control

Once delivered, please record all changes made to this document.

Date	Version	Author	Section	Amendment
2006				Beta version of document made available
10/23/07	1.0	M. Eaton		Official Version 1.0 released
11/15/07	1.01	D. Bruton		Editing and minor formatting
08/11/08	1.02	D. Bruton		Formatting to be consistent with other EMS documents and inclusion of new Unica document

1.2 Master Documents

Document ID	Source	Title
	Unica	Page Tag User's Guide
	Unica	NetInsight User's Guide

1.3 Operational Downloads

Source	**Title**	**Location**
ESDIS	Page Tag Implementation Code and Examples	http://ems.eos.nasa.gov/examples/ntpagetag.zip

This page is intentionally left blank.

2 Introduction

The ESDIS Metrics System (EMS) uses the EMS NetInsight® page tagging script to collect Web usage information from data provider Web sites. Data are collected by “tagging” Web pages with JavaScript, causing a visitor’s browser to send information about the visit to the EMS page tag server on ws1.ems.eosdis.nasa.gov. This information, which appears in the page tag server log as a request for the page tagging image, is then imported by EMS for analysis. This document will help you:

- Install and configure the components in the EMS NetInsight page tagging script.
- Resolve some general and specific questions you may have about page tagging and on-page event tagging.
- Set required and optional page tagging variables and call page tagging functions.

We recommend that you read the entire document before you install and configure the EMS NetInsight page tagging script.

2.1 How Page Tagging Works

Page tagging works by causing your Web server to request a small transparent image file from the page tag server for every activity you want to track. The activity could be a request for a page, the addition of an item to a shopping cart, a click on a link, an interaction with a Rich Internet Application (RIA), or any other event on your site. Because the image file is transparent, your site users will never notice it was served, but the image file request will be logged in the page tag server’s log file and can be analyzed using NetInsight.

Page tag image requests by default contain the following information:

- IP address
- date/time
- page with query string
- referrer
- user agent
- cookie

This information will appear in NetInsight reports. You can also add other information to the page tag requests and configure NetInsight parameters to report on that information. You can tag all of your Web site’s pages or only some of them. You can use page tagging in conjunction with or instead of analyzing your Web server’s log files.

2.2 When to Use Page Tagging

Here are some of the reasons you would use page tagging:

- You do not have access to the log files of your Web site’s server, and you want to analyze traffic on the site.

- You want to analyze additional information (such as screen resolution) about your site's visitors that is not transferred during a page request but is transferred during a page tag request.
- You want to track events, specifically actions on your Web site other than loading a page. Events include but are not limited to the following: changing a field on a form, selecting an option in a drop-down list box, or submitting a form. Events are plentiful in RIAs such as Flash or AJAX applications.

3 Installation

3.1 EMS NetInsight Page Tagging Script

The EMS NetInsight page tagging script is bundled in a ZIP file named ntpagetag.zip, which you can download from http://ems.eos.nasa.gov/examples/ntpagetag.zip.

3.2 Browsers with JavaScript Disabled

In order for the page tag script to work, the visitor's browser must have JavaScript enabled. The Unica page tag contains an alternative page tag image request that is sent when the browser does not have JavaScript enabled. The presence of this request in the page tag server's log files tells NetInsight that the visitor's browser did not allow page tags to be used.

3.3 EMS Page Tagging Components

The ntpagetag.zip file contains the following files:

- ntpagetag.js: a JavaScript file that enables EMS NetInsight page tagging on a Web site
- sample.html: a file that contains sample page tags, including event tags
- power.pdf: a file describing how to use NetInsight reports
- recipecards.pdf: a file showing Web analytic recipes

3.4 Installing the Page Tagging Script

1. Unzip ntpagetag.zip and extract the files to a directory of your choosing.
2. Ensure that http://ws1.ems.eosdis.nasa.gov/images/ntpagetag.gif, the remote image, is accessible. To do this, try to load the image by entering its URL in a browser. Because ntpagetag.gif is a one-pixel transparent image, you will see a blank page if it loads successfully. You can ensure that the image is being served by requesting that EMS staff (ems-help@lists.nasa.gov) check the page tag Web server log file for your request.
3. Open ntpagetag.js in a text editor and do the following:
   - Find the line "*`var NTPT_IMGSRC = '/images/ntpagetag.gif';`*". Ensure that the URL specified in the definition is http://ws1.ems.eosdis.nasa.gov/images/ntpagetag.gif.
   - Find the line following the "*var NTPT_FLDS = new Object();*" line. Change the properties of the NTPT_FLDS variable from "true" to "false" to change which fields are included in the page tag.
4. Save any changes to ntpagetag.js.
5. Copy ntpagetag.js to the Web site on which you want to implement EMS NetInsight page tagging. For example, if your Web site is located at www.foo.com, you could copy the file to http://www.foo.com/scripts/ntpagetag.js. For details, see **"Resolving "Cross-Site Scripting" Warnings."**

   Note: Ensure that ntpagetag.js is publicly accessible on the Web server.

6. Open sample.html in a text editor and do the following:
   a. Find the EMS NetInsight page tag, which consists of these lines:
      ```
      <!-- BEGIN: EMS NetInsight Page Tag -->
      <!-- Copyright 2004 Sane Solutions, LLC. All rights reserved. -->
      <script language="JavaScript" src= "/scripts/ntpagetag.js"></script>
      <noscript>
      <img src="/images/ntpagetag.gif?js=0" height="1" width="1" border="0"
      hspace="0" vspace="0" alt="">
      </noscript>
      <!-- END: EMS NetInsight® Page Tag -->
      ```
   b. If necessary, change the JavaScript location in the “src” attribute to match the location (either relative or absolute) of the ntpagetag.js file on the Web server.
   c. If necessary, change the img element’s “src” attribute to point to the location of the ntpagetag.gif image on the EMS Web server. Find the line “var NTPT_IMGSRC = '/images/ntpagetag.gif';”. Ensure that the URL specified in the definition is http://ws1.ems.eosdis.nasa.gov/images/ntpagetag.gif.
   d. Select and copy the EMS NetInsight page tag as shown in Step 6a above.
   e. Save any changes to sample.html.
7. To tag the pages on your Web site, you can use any of the following methods:
   a. Use Server Side Includes, to include the page tag script in Step 6a in your Web pages.
   b. Use mod_layout, if you are using Apache (available from http://tangent.org/).
   c. Use the content management system to imbed a file containing the page tag into your template.
   d. Use any other effective method that suits your needs
8. Copy your tagged HTML pages to your Web server. The views of the pages will now be logged.
9. Complete a page tag manifest worksheet shown in **Appendix A** and send it to ems-help@lists.nasa.gov.

3.5 Resolving “Cross-Site Scripting” Warnings

A “cross-site scripting” warning occurs when a client-side script (such as the ntpagetag.js JavaScript file) is served from a different domain than the page that includes it. In this situation, you may get a warning from your browser, indicating a cross-site scripting attempt, and the script may not execute. For this reason, the ntpagetag.js file should be located on the same Web server as the Web pages you are tagging.

Note: The ntpagetag.gif file located on the EMS server will not cause a cross-site scripting warning.

3.6 Tagging Secure Pages without Generating Browser Warnings

Sometimes browsers warn users if both secure and insecure content exist on a page. For this reason, it is important not to issue an insecure (http) tag request when serving a secure (https) page.

When a page is going to be served securely, which means that it can be accessed at a location beginning with https://, the ntpagetag.js script tag's "src" attribute should not be absolute unless it also begins with https://.

In addition, if the NTPT_IMGSRC variable in the header of the ntpagetag.js file is set to an insecure location, then the NTPT_HTTPSIMGSRC variable (also located in the header of the ntpagetag.js file) should be set to a secure location.

This page is intentionally left blank.

4 Basic Page Tagging

Basic page tags capture details about the current page when the page loads. You do not need custom scripting to use basic page tags.

4.1 Before You Tag Pages

Before you tag the pages you want to track, you need to install and configure the page tag and page tag script.

4.2 Tagging Pages

1. Paste the Unica page tag into each page you want to track.

   If you want to track the entire site, paste the page tag into the master template or a global include file, such as a footer.

2. Customize the page tagging as needed for each page, using the optional page-specific variables.
3. Copy your tagged HTML pages or the updated master template or include file to the Web server.

If you do not need to customize page tags on a case-by-case basis or use page tags to track events or order activity on your site, you are finished configuring your Web site to use page tags.

4.3 Customizing Page Tags

You can customize the page tag request on a page-by-page basis by using optional page-specific variables, such as NTPT_PGEXTRA. Refer to Chapter 6, "Page Tagging Variables and Functions" for more information.

4.4 Verifying That the Page Tags Are Working

You can verify that the page tags are working by ensuring that the page tag server's log files contain requests for the Unica page tag image (ntpagetag.gif). The log files should contain one line for each request for the ntpagetag.gif image similar to the following example:

```
192.168.0.64 - - [25/Jul/2004:07:30:49 -0400]
"GET /images/ntpagetag.gif?js=1&ts=1089199849489.408&lc=http%3A
//testserver/sample.html&rf=http%3A
//testserver/&rs=1280x1024&cd=24&ln=en&tz=GMT%20-04%3A00&jv=0
HTTP/1.1" 200 85 "http://testserver/sample.html"
"Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.3.1) Gecko/20030425".
```

4.5 How Netinsight Extracts Information from the Page Tag Request

4.5.1 Page Tagging Options in NetInsight

The page tagging options in NetInsight determine how NetInsight uses the data in the page tag requests.

- If the page tag options are set to Not used, NetInsight will not use any of the information in the page tag request, although the requests will still count as hits.
- If the page tag options are set to Used, only to augment log files with additional data, NetInsight will extract and report on the visit-level information in the page tag request. Page tag requests will count as hits but not page views. Events will count as events.
- If the page tag options are set to any other value, NetInsight will count page tag requests as both hits and page views and will extract and report on all the information in the page tag request. Events will count as events.
- If the page tag request includes pv= or ev=, those values will override the page tag option's default behavior regarding page views.

4.5.2 About the Query String

The information NetInsight analyzes is stored in the page tag request's query string in the form of name-value pairs. NetInsight recognizes some names by default and will store and analyze the values. If you want to analyze more information, you can add name-value pairs to the page tag and then create a parameter in NetInsight so that the information will be stored in the database and reported on in NetInsight.

By default, the page tag query string looks like this:

```
js=1&ts=1089199849489.408&lc=http%3A//testserver/
sample.html&rf=http%3A//testserver/
&rs=1280x1024&cd=24&ln=en&tz=GMT%20-04%3A00&jv=0
```

5 Advanced Page Tagging

5.1 Tagging Events

5.1.1 Before You Tag Events

Before you tag the events you want to track, you need to install and configure the page tag script and page tag.

5.1.2 About Events

An event is any on-page action other than loading a page. Events include but are not limited to the following:

- Changing a field on a form
- Selecting an option in a drop-down list box
- Submitting a form

Events are plentiful in RIAs such as Flash or AJAX applications.

5.1.3 About Event Tagging

Event tagging is a special form of page tagging. Unlike basic page tags, event tags capture details about a visitor's interaction with a page after the page loads. Event tagging requires custom scripting to call the event tagging API.

The main functions in the event tagging API are ntptEventTag, ntptLinkTag, and ntptSubmitTag. Every event you want to track must call one of these functions. Events may also call ntptAddPair and ntptDropPair in conjunction with one of the main functions.

5.1.4 Tagging JavaScript or AJAX Events

1. Paste the page tag into each page that contains an event you want to track.
2. Customize the page tagging as needed for each page, using the optional page-specific variables described in Chapter 6, "Page Tagging Variables and Functions."
3. Create a JavaScript event handler for every event you want to track. Each event handler should call the appropriate event-tagging functions also described in Chapter 6, "Page Tagging Variables and Functions."
4. Copy your tagged HTML pages to the Web server or your content management system.

The example below shows three different ways to call the event-tagging functions. The first instance uses the onchange attribute of the <input> HTML element to call the ntptEventTag function directly. The second instance uses the onchange attribute of the <input> HTML element to call the ntptEventTag function and pass additional information. In this example, the value of the text box is passed to the page tag request. The final instance uses a custom event handler named MyEventHandler to call the ntptAddPair and ntptEventTag functions in a JavaScript function.

```
<form>
<input type="checkbox" name="mybox" onchange="ntptEventTag();">
<input type="text" name="mytext"
onchange="ntptEventTag('mytext_changed=1&mytext=" +
escape( this.value ) + "' );">
<script language="JavaScript">
function MyEventHandler()
{
ntptAddPair( "color", "red" );
ntptEventTag( "ev=myevent" );
}
</script>
</form>
```

5.1.5 Tagging Events in Flash Presentations

1. Create an ActionScript event handler for every event you want to track. Each event handler should call the appropriate JavaScript event-tagging functions described in Chapter 6, "Page Tagging Variables and Functions."
2. Publish your Flash using the HTML template "Flash with FSCommand," which adds Adobe-supplied JavaScript to the HTML container file to send events to the Page Tagging JavaScript API.

One of the functions Adobe adds to your HTML is DoFSCommand. You must modify the DoFSCommand function to enable it to call the JavaScript event-tagging functions. (See Adobe's Flash documentation on Flash-to-JavaScript communication for details.) Replace the sample code in the DoFSCommand function with the following JavaScript code:

```
if (null == args)
return;
var tmpargs = args.split(",");
if ( command == "ntptEventTag" )
{
(0 == tmpargs[0].length) ? ntptEventTag() : ntptEventTag(
tmpargs[0] );
}
else if ( command == "ntptAddPair" )
{
if ( 2 != tmpargs.length )
return;
else
ntptAddPair( tmpargs[0], tmpargs[1] );
}
else if ( command == "ntptDropPair" )
{
if ( 1 != tmpargs.length )
return;
else
ntptDropPair( tmpargs[0] );
}
```

3. Paste the page tag into the page that references the Flash presentation.
4. Customize the page tagging for the page as needed using the optional page-specific variables described in Chapter 6, “Page Tagging Variables and Functions.” The example below shows a custom ActionScript event handler named PickColor. The first line assigns the custom function to the onPress event handler of the object named ColorButton. The PickColor function first calls another user-defined function named UpdateCarPhoto and then calls the event-tagging functions using fscommands.

```
ColorButton.onPress = PickColor;
function PickColor()
{
UpdateCarPhoto( ColorButton.value );
fscommand ( "ntptAddPair", "color=" + ColorButton.value );
fscommand( "ntptEventTag", "ev=pickcolor" );
}
```

5.1.6 Ensuring That Link and Submit Page Tag Requests Are Submitted before the Page Unloads

Following a link or submitting a form ultimately leads to the unloading of the page that contains the link or the form. To ensure that the event tag request is sent before the page is unloaded, the page tag script introduces a small delay when tagging links and submissions. This delay is the maximum amount of time that will elapse before the page is unloaded; if the event tag request returns before the specified time has elapsed, the page will be unloaded immediately without waiting for the rest of the time to elapse. You can configure this delay globally by setting the NTPT_MAXTAGWAIT variable. You can adjust the wait for an individual link or submission by using the maxtagwait argument for ntptLinkTag or ntptSubmitTag.

The default wait is one second, which is usually indiscernible to the user and sufficient to capture all tagged links and form submissions effectively.

5.1.7 Marking Link Tags as Links to an External Site

In NetInsight, links to external sites are analyzed in the Link Summary. You can mark a link page tag request as a link to an external site by including the name-value pair “lk=1” in the page tag request’s query string. You can pass the lk=1 name-value pair into the query string by including it in the querymod argument for ntptLinkTag or by using ntptAddPair to add it before you call ntptLinkTag.

5.1.8 Ensuring That an Event Does Not Count as a Page View

There are two ways to ensure that NetInsight does not count an event as a page view:

- Set the field-value pair ev=*event type* using the query modifier of the ntptEventTag function, ntptLinkTag function, or ntptSubmitTag function. NetInsight will count the event as an event and not as a page view, and the event will automatically be included in the Event Summary. Additional field-value pairs from this page tag can also be parsed in NetInsight as parameters related to the event.
- Set the field-value pair pv=0 using the NTPT_PGEXTRA variable, the ntptAddPair function, or the query modifier argument of the ntptEventTag function, ntptLinkTag function, or ntptSubmitTag function. NetInsight will not count the event as a page view or an event.

5.2 Tagging Order Activity

5.2.1 About Order Activity

Order activity is selecting and purchasing products. It includes the following actions:

- Viewing products
- Adding products to a shopping cart
- Removing products from a shopping cart
- Completing the checkout process

The best practice for tagging order activity involves a combination of page tags and event tags.

5.2.2 Tagging Product Views

When you serve a page of products for viewing, you should use your ordering platform or content management system to pass the Order ID of the products on the page into the rtv field in the

NTPT_PGEXTRA variable.

Example

```
NTPT_PGEXTRA="rtv=FY210;SX042;EP360" ;
```

However, if the page also contains event tags, you will need to set the rtv field to empty before the event tag. Otherwise, the page tag server will log one product view when the page is initially displayed and another product view when the event tag is sent.

```
NTPT_PGEXTRA="rtv="
```

5.2.3 Tagging Additions to the Shopping Cart

You should call ntptEventTag from the JavaScript or ActionScript cart handler function that enables a visitor to add an item to the cart. Pass the Order ID and quantity for each product added to the cart into the rta name-value pair in the querymod argument of ntptEventTag. You can pass information on multiple products in a single call. NetInsight interprets the first two values as the Order ID and quantity of the first product, the next two values as the Order ID and quantity of the second product, etc.

5.2.4 Tagging Removals from the Shopping Cart

You should call ntptEventTag from the JavaScript or ActionScript cart handler function that enables a visitor to remove an item from the cart. Pass the Order ID and quantity for each product removed from the cart into the rtr name-value pair in the querymod argument of ntptEventTag. You can pass information on multiple products in a single call. NetInsight interprets the first two values as the Order ID and quantity of the first product, the next two values as the Order ID and quantity of the second product, etc.. The unit price should be a float value, without any currency signs. Including currency signs will result in a unit price of $0 within NetInsight. You should pass either pv=0 or ev=*event type* to avoid inflating page views.

Example:

```
ntptAddPair( "rtr", sSku + ";" + iQuantity + ";" + fPrice );
ntptEventTag( "pv=0" );
```

5.2.5 Tagging the Checkout Process

When you serve the thank you page at the end of the checkout process, you should use your ordering platform or content management system to pass the Order ID and quantity of each product added into the shopping cart into the rtc field of the NTPT_PGEXTRA variable for the order into the rtt field.

5.3 Tagging Visit Cost

5.3.1 About Visit Cost

Visit cost is the amount of development or promotion person-hour cost spent to direct a visitor to your site for a particular visit.

5.3.2 Including Visit Cost Data in a Page Tag Image Request

You can include visit cost data in a page tag image request by passing the visit cost data using the name-value pair vc=*value*.

This page is intentionally left blank.

6 Page Tagging Variables and Functions

The following JavaScript variables and functions, described later in this chapter, are available for the pages you are tagging:

- Required variables in ntpagetag.js
- Optional variables in ntpagetag.js
- Optional page-specific variables
- User-callable page tagging functions (event tagging)

6.1 Required Variables in ntpagetag.js

The following three variables, set in ntpagetag.js, are required.

6.1.1 NTPT_IMGSRC

A string variable that contains the URL of the page tag image. If the page tag image and the page tag script are on the same Web server as the pages you are tagging, you can omit the protocol and server. However, if the page tag script is on a different server than the Web pages you are tracking, you must specify the fully qualified URL to the page tag image, even if the page tag image and page tag script are located on the same server. For example:

Syntax
```
var NTPT_IMGSRC = ‘URL’;
```
Example
```
var NTPT_IMGSRC =
'http://ws1.ems.eosdis.nasa.gov/images/ntpagetag.gif'
```

6.1.2 NTPT_FIELDS

An object variable that specifies the fields that will be included in the default page tag. Fields set to true will be included in the page tag. Fields set to false will not be included. In most cases, you should use the default field settings. Every page tag should include the lc field.

These fields are defined as follows:

```
var NTPT_FLDS = new Object();
NTPT_FLDS.lc = true; // Document location
NTPT_FLDS.rf = true; // Document referrer
NTPT_FLDS.rs = true; // User's screen resolution
NTPT_FLDS.cd = true; // User's color depth
NTPT_FLDS.ln = true; // Browser language
NTPT_FLDS.tz = true; // User's time zone
NTPT_FLDS.jv = true; // Browser's Java support
```

You can set the field attributes to *true* or *false*. EMS strongly recommends that you set all values to the default found in sample.html.

A value of *true* will include the field in the page tag.

A value of *false* will omit the field from the page tag.

6.1.3 NTPT_MAXTAGWAIT

The numeric value for this variable defines the maximum number of seconds that a call to ntptLinkTag() or ntptSubmitTag() will wait for the page tag request to be sent. Because the page tag request may come back before this timeout, this number should be considered a "worst case" delay for link tags and submit tags.

Valid values are:

- Any number greater than zero—Positive numbers indicate the number of seconds to wait before following the link or submission. Note that setting the NTPT_MAXTAGWAIT variable to 1.0 (one second, the default) is usually indiscernible to the user, and is usually sufficient to capture all tagged links and form submissions effectively.

  Sometimes browsers do not execute the page tag's JavaScript code if the page is unloading at the same time that the page tag is executing. In such instances, you can use a value greater than 0 to ensure that the page tag is executed before the page is unloaded.

  Note: You can override the default value of 1.0 second for any given call to ntptLinkTag() and ntptSubmitTag() by passing in a value for the "maxtagwait" argument.
- -1 — A value of -1 (minus 1) indicates that no timeout should be used. This value should be used with caution, as some browsers require a small delay to tag an event that unloads the page, such as clicking on a link or submitting a form.

Example:

```
var NTPT_MAXTAGWAIT = 1; // one second
var NTPT_MAXTAGWAIT = 2.5; // two and a half seconds
var NTPT_MAXTAGWAIT = 0.1; // a tenth of a second
var NTPT_MAXTAGWAIT = -1; // no delay
```

6.2 Optional Variables in ntpagetag.js

The following three variables, set in ntpagetag.js, are optional.

6.2.1 NTPT_HTTPSIMGSRC

This string variable, like NTPT_IMGSRC, can contain a URL for a page tag image. If NTPT_HTTPSIMGSRC is set and is not empty, use this variable when you accessed the page being tagged using the "https:" protocol.

Syntax
```
var NTPT_HTTPSIMGSRC = 'image';
```
Example
```
var NTPT_HTTPSIMGSRC = 'https://mysite.com/images/ntpagetag.gif';
```

6.2.2 NTPT_GLBLEXTRA

This string variable is the first "query modifier" to be applied to a page tag. A query modifier is a string that can either be empty or contain key-value pairs that will act to modify the initial page tag query string.

If a key is not present in the initial query string, the key-value pair from the query modifier will be appended to the resultant query string.

If a key from the query modifier is already present in the page tag's initial query string, the value of that key will be replaced by the new value in the query modifier. If that modifier value is empty, then the key-value pair will be dropped from the resultant query string.

The NTPT_GLBLEXTRA query modifier is applied to every page tag. Use it to modify every page tag on every page that includes ntpagetag.js.

Syntax
```
var NTPT_GLBLEXTRA = 'key=value';
```
Example
```
// Append the 'sitetheme=blue' pair to every page tag.
var NTPT_GLBLEXTRA = 'sitetheme=blue';
```

6.2.3 *NTPT_GLBLREFTOP*

You can set this variable to *true* or *false*. If the variable is *true*, the referrer (the "rf" field) is retrieved from the top (that is, the most containing) frame of the current page. Otherwise, the referrer is retrieved from the current page. If you do not set this variable, it defaults to *false*.

Note: The NTPT_PGREFTOP variable, if set, overrides this variable's value.

Syntax
```
var NTPT_GLBLREFTOP = [true,false];
```
Example
```
var NTPT_GLBLREFTOP = true;
```

6.3 Optional Page-Specific Variables

The page-specific variables described on the following pages are optional. Note that you must define the page-specific variables before the NetInsight page tag is loaded. Because the <head> element for the page is processed first, it is often a good place to define the page-specific variables, as shown in the following HTML example:

```
<head>
<!-- BEGIN: Use this section to set page specific variables for the
NetInsight® Page Tag -->
<script language="JavaScript">
var NTPT_PGEXTRA = 'pagetheme=blue&rf=';
var NTPT_PGREFTOP = false;
var NTPT_NOINITIALTAG = false;
</script>
<!-- END -->
<!-- BEGIN: NetInsight® Page Tag -->
<!-- Copyright 2004 Sane Solutions, LLC. All rights reserved. -->
<script language="JavaScript" src="/scripts/ntpagetag.js"></script>
<noscript>
<img src="/images/ntpagetag.gif?js=0" height="1" width="1" border=
"0" hspace="0" vspace="0" alt="">
</noscript>
<!-- END: -->
</head>
```

6.3.1 NTPT_PGEXTRA

This variable is the second “query modifier” to be applied to a page tag. Because the NTPT_PGEXTRA modifier is applied after NTPT_GLBLEXTRA (the first query modifier), it can override key-value pairs in either the initial query string or NTPT_GLBLEXTRA. For a description of query modifiers, see NTPT_GLBLEXTRA in Section 6.2, Optional Variables in ntpagetag.js.

Note: The “sc” (status code) field provides NetInsight with a status code it can use instead of the status code associated with the request. Defining the “sc” field in NTPT_PGEXTRA is especially useful for logging error requests. For example, you can assign the “sc” field a value greater than or equal to 400 in the page tag on your custom error page so that NetInsight will treat that request as an error.

Syntax

```
var NTPT_PGEXTRA = 'field=value&field=value&field=value[...]';
```

Example

```
// Append the 'pagetheme=red' pair. Drop the 'rf' field.
var NTPT_PGEXTRA = 'pagetheme=blue&rf=';
// Treat this page as a 404 (Not Found) error.
var NTPT_PGEXTRA = 'sc=404';
```

Note: This query modifier is not applied to page tags generated by a call to ntptLinkTag(), as the purpose of a link tag is to simulate a page tag on another page. To modify the query string of a link tag, you can use the “querymod” argument to that function.

To ensure that a page tag appears as a separate page view in NetInsight®, you can set the “pv” field to a value of 1. This is true even when NetInsight is configured to augment only log files with additional data, which normally discards the tag when tracking page views.

To discard a request that would have been counted, you can set the “pv” field to a value of 0. This is useful when using event tags; using “pv=0” when firing an event tag enables parameters to be parsed from the request and associated with the corresponding visit without causing the request to be counted as a page view.

```
var NTPT_PGEXTRA=’GeoExtentLat=23.5&GeocExtentLng=96.3
&SpectralBands=1,2,3,4&DateRange=12/01/2005-12/31/2005
&DataType=MOD12&OrigToSubsetVolRatio=.10’
```

Note: The example above has six parameter value pairs that will be captured by the EMS using the NTPT_PGEXTRA variable in the page tag JavaScript. The total number of characters that can be placed in the NTPT_PGEXTRA variable is limited to 1,500 including parameter names, parameter values, special symbols and delimiters.

The format of the NTPT_PGEXTRA variable uses the convention shown in the example below:

```
//var NTPT_PGEXTRA =
‘parameter1=value1&parameter2=value2&parameterN=valueN’;
```

Figure 1: Format of NTPT_PGEXTRA Use in POST Example

Figure 2 below illustrates the JavaScript to be inserted into POST method-generated Web pages. Note that the only difference between the generic page tag and this example is the var NTPT_PGEXTRA line. You can locate complete page tag installation and configuration instructions at http://ems.eos.nasa.gov/documents.

```
<!-- BEGIN: EMS Page Tag Notes -->
<!-- Some content Copyrighted 2004 Sane Solutions, LLC.  All rights
reserved. -->
<!-- Modified 02/07/2005 kjm -->
<!-- Page tag script to transfer data from POSTs-->
<!-- This information should be placed directly before the closing
BODY tag -->
<!-- YOU MUST CHANGE src="/scripts/ntpagetag.js" to comply with your
local implementation -->
<!--Uncomment the 'var NTPT_PGEXTRA=""' line and insert application
specific parameter value pairs from POSTs -->
<!-- END: EMS Page Tag Notes -->
<!-- BEGIN: EMS Page Tag -->
<script language="JavaScript">
```

var NTPT_PGEXTRA='GeoExtentLat=23.5&GeocExtentLng=-96.3& SpectralBands=1-4&DateRange=12/01/2005-12/31/2005&DataType=MOD12&OrigToSubsetVolRatio=.10';

```
// var NTPT_PGREFTOP = false;
// var NTPT_NOINITIALTAG = false;
</script>
<script language="JavaScript" src="/scripts/ntpagetag.js"></script>
<!-- END: EMS Page Tag -->


<!-- BEGIN: EMS Page Tag NOSCRIPT -->
<noscript>
<img src="http://ws1.ems.eosdis.nasa.gov/images/ntpagetag.gif?js=0"
height="1" width="1" border="0" hspace="0" vspace="0" alt="">
</noscript>
<!-- END: EMS Page Tag No Script -->
```

Figure 2: Example of Page Tag for Capturing POST Method Parameters

6.3.2 NTPT_PGLREFTOP

This variable serves the same purpose as NTPT_GLBLREFTOP. It specifies whether the referrer should be retrieved from the top frame or from the current page (the default). If you set NTPT_PGREFTOP, it will override NTPT_GLBLREFTOP. For details on NTPT_GLBLREFTOP, refer to "NTPT_GLBLREFTOP."

Syntax
```
var NTPT_PGREFTOP = [true,false];
```
Example
```
var NTPT_PGREFTOP = true;
```

6.3.3 NTPT_NOINITIALTAG

If you set NTPT_NOINITIALTAG to true, the initial page tag for the current page will be suppressed and only event tags will fire on the page.

Syntax
```
var NTPT_NOINITIALTAG = [true,false];
```
Example
```
var NTPT_NOINITIALTAG = true;
```

6.4 User-Callable Page Tagging Functions (Event Tagging)

The following pages describe the user-callable page tagging functions. Examples are available in the sample.html (one of the files in the ntpagetag.zip file).

6.4.1 ntptAddPair

The "key" and "value" arguments constitute a key-value pair to add to the working query string of the next event tag to fire. If the key already exists in the working query string, it will be replaced. If the "value" argument is empty (' '), the pair will be dropped.

Call this function to modify the working query string, prior to firing an event tag.

Note: The working query string is reset to the initial query string after every event tag fires.

Argument Description

Key: The name of a name-value pair to add to the next event tag to fire.

Value: The value of a name-value pair to add to the next event tag to fire.

Syntax

```
ntptAddPair( key, value )
```

Example

```
ntptAddPair( "color", "red" );
```

6.4.2 ntptDropPair

This function drops the pair matching the "key" argument from the working query string. Call this function to modify the working query string, prior to firing an event tag.

Note: Calling ntptDropPair('mykey') is equivalent to calling ntptAddPair('mykey', ' ').

Argument Description

Key: The name of a name-value pair to drop from the next event tag to fire.

Syntax

```
ntptDropPair ( key )
```

Example

```
ntptDropPair( tmpargs[0] );
```

6.4.3 ntptEventTag

This function fires an event tag using the working query string. This function should be called from a document element's event handler.

Modifications to the working query string can be made by the optional "querymod" argument, which is a query modifier like NTPT_GLBLEXTRA and NTPT_PGEXTRA.

Because the "querymod" query modifier will be the last modifier applied to the working query string before the page tag is requested, it may replace or drop any existing key-value pairs in the working query string.

The working query string is reset to the initial query string after the event tag is requested. For example:

```
<input type="text" name="mytext" onchange="ntptEventTag(
'mytext_changed=1&mytext=" + escape( this.value ) + "' );">
```

Argument Description

querymod A query modifier for the event tag. It modifies the working query string for the page tag.

Syntax
```
ntptEventTag ( [querymod] )
```
Example
```
ntptEventTag( "ev=pickcolor" );
```

6.4.4 ntptLinkTag

This function is used to tag links that would otherwise not be accessible to page tagging. These links include downloads and non-HTML pages. This function should always be called from the "onclick" attribute of a link.

When calling ntptLinkTag(), you should always return the value of the function to the "onclick" handler. That is, the "onclick" attribute of the anchor should include "returnntptLinkTag(this);" and should not omit the "return" before the ntptLinkTag() call.

The arguments are:

- Linkobj: This argument should always be the keyword "this" so that the function can follow the link after tagging it.
- querymod This optional argument is a query modifier for the link tag. It is used, like the optional querymod argument to ntptEventTag(), to modify the working query string just prior to the page tag request.
- maxtagwait This optional argument overrides the value of the global

NTPT_MAXTAGWAIT variable for this link tag only. It specifies the maximum number of seconds to wait in order to ensure that the link tag request was fired. The page will not unload, and the link will not be followed until either the specified number of seconds have elapsed or the tag request returns, whichever happens first. Passing a "maxtagwait" value of -1 will cause no delay in following the link.

Note: Link tags are not modified by the NTPT_PGEXTRA variable.

Example:

```
<a href="mymovie.mpeg" onclick="return ntptLinkTag( this );">
```

Argument Description

linkobj A link object. The keyword "this" tells the function to follow the link after tagging it. querymod A query modifier for the link tag. It modifies the working query string for the page tag. maxtagwait The maximum number of seconds that thecall will wait before following the link. This overrides the global wait time specified by NTPT_MAXTAGWAIT.

Syntax

```
ntptLinkTag ( linkobj [, querymod [, maxtagwait]] )
```

Example
```
onclick="return ntptLinkTag( this );"
```

6.4.5 ntptSubmitTag

This function tags form submissions. It should always be called from the “onsubmit” attribute of a form, such as:

```
<form name="myform" onsubmit="return ntptSubmitTag( this );">
```

When calling ntptSubmitTag(), you should always return the value of the function to the “onsubmit” handler. That is, the “onsubmit” attribute of the form should include “return ntptSubmitTag(this);” and should not omit the “return” before the ntptSubmitTag() call.

The arguments are:

- formobj This argument should always be the keyword “this” so that the function can submit the form after tagging it.
- querymod This optional argument is a query modifier for the submit tag. It is used, like the optional “querymod” argument to ntptEventTag() and ntptLinkTag(), to modify the working query string just prior to the tag request.
- maxtagwait This optional argument overrides the value of the global

NTPT_MAXTAGWAIT variable, for this submit tag only. Like the “maxtagwait” argument to ntptLinkTag(), it specifies the maximum number of seconds to wait in order to ensure that the submit tag request was fired. The page will not unload and the form will not be submitted until either “maxtagwait” seconds have elapsed, or the tag request returns, whichever happens first. Passing a “maxtagwait” value of -1 will cause no delay in submitting the form.

Argument Description

maxtagwait The maximum number of seconds that the call will wait before submitting the form. This overrides the global wait time specified by NTPT_MAXTAGWAIT.

Syntax

```
ntptSubmitTag( formobj [, querymod [, maxtagwait]] )
```

Example

```
return ntptSubmitTag( document.myform );
```

7 Field NetInsight Recognizes Automatically

The following fields do not appear in the page tag request by default, but if you add them, NetInsight will recognize them automatically and report on their values. You do not need to create parameters in order for the information from these fields to appear in your NetInsight reports.

You can add these fields to the page tag request by using the NTPT_PGEXTRA variable, the ntptAddPair function, or the querymod argument of the ntptEventTag, ntptLinkTag, or ntptSubmitTag functions.

Field	Description	Possible values
pv	Code that determines whether or not NetInsight counts the page tag request as a page view in NetInsight. A value of 0 does not count the request as a view. A value of 1 counts the request as a view.	0, 1
sc	Status of the request (HTTP status code)	Number
rtv	Products viewed on the current page	List of product SKUs separated by semi-colons
rta	Products added to the shopping cart on the current page	List of product SKUs, quantities, and values separated by semi-colons
rtr	Products removed from the shopping cart on the current page	List of product SKUs, quantities, and values separated by semi-colons
rtt	Revenue associated with the current page	Number
rti	Order ID associated with the current page	String
rtc	Products purchased	List of product SKUs, quantities, and values separated by semi-colons
vc	Visit cost (the amount of money spent to drive a visitor to your site for this visit).	Number
lk	Code that determines whether or not NetInsight counts the page tag request as a link to an external site. A value of 0 does not count the request as an external link. A value of 1 counts the request as a link to an external site.	0, 1
ev	Type of event	Any text value

This page is intentionally left blank.

8 Frequently Asked Questions

This section lists some general and specific questions you may have about the EMS NetInsight® page tagging script. The questions are divided into six categories:

8.1 General

What is an EMS NetInsight page tag image?

The page tag image is a one-pixel transparent image that resides on the EMS server.

How does page tagging analysis compare to conventional Web server log analysis?

Page tagging is a method of collecting data about page views and events on your Web site that does not require access to your site's log files.

Differences exist in both the type of information you can collect as well as the method you use to collect that information. Page tagging relies on the visitor's browser to send information about his or her visit; conventional Web server logs do not rely on the browser in this way. Conversely, page tagging solutions may provide more visitor information than conventional Web server logs.

What information does a page tag image transmit?

By default it contains the following information:

• IP address

• date/time

• page URL with query string

• referrer

• user agent

• cookie (if used)

This information will appear in NetInsight reports. You can also add other information to the page tag requests and configure NetInsight parameters to report on that information.

Do I have to tag my product-ordering client applications?

No, capturing information from Web application query strings or POST method-generated pages is optional.

8.2 Configuring Page Tags

What additional information can the EMS page tag provide?

The following fields can be included or excluded in the page tag by using the NTPT_FLDS variable. For more information, see "NTPT_FLDS" in **Page Tagging Variables and Functions**. ***Generally, you should not change any of these fields.*** This information is included in case there are any problems with your deployment.

Field	Description	Example
cd	Color depth of the user's screen	cd=24
jv	User's browser support for Java	jv=0
lc	Document location (that is, the page that you are tagging)	lc=http%3A//testserver/sample.html
ln	Language encoding that the user's browser is configured to use	ln=en
rf	Referrer for the request	rf=http%3A//testserver/
rs	User's screen resolution, in the format *width*x*height*	rs=1280x1024
tz	Time zone of the user's server	tz=GMT%20-04%3A00

You can also set these optional page-specific fields for the page tag by using the NTPT_PGEXTRA variable. For more information, see "**NTPT_PGEXTRA**" in **Optional Page-Specific Variables.**

Field	Description	Example
pv	Counts a page tag request as a separate page view in EMS NetInsight® if it would normally have been discarded, or discards a request that would have been counted as a page view	pv=0
sc	Status of the request (HTTP status code)	sc=404

How do I include or exclude information from the page tags on my Web site?

You can include or exclude the available fields by setting the relevant NTPT_FLDS properties in the header of the ntpagetag.js file. For example, to include the user's screen resolution in the page tag, set the NTPT_FLDS.rs property to "true." To exclude this field, set the property to "false." ***Recommended settings for these variables are already specified in sample.html.*** For details on NTPT_FLDS, see "NTPT_FLDS" in **Optional Page-Specific Variables**.

How do I control what referrer information is sent for a Web site with frames?

The default behavior is to report each frame's referrer as the page that opened the frame. You can instead report each frame's highest-level parent's referrer in its referrer field in the page tag. You can make this change in one of two ways:

To change the behavior for all pages, you can set the NTPT_GLBLREFTOP variable to "true" in the header of the ntpagetag.js file. For details on NTPT_GLBLREFTOP, see "NTPT_GLBLREFTOP" in **Optional Variables in ntpagetag.js**.

To change the behavior for individual pages, you can set the NTPT_PGREFTOP variable in the page. NTPT_PGREFTOP, if set, overrides any value of NTPT_GLBLREFTOP. For details on NTPT_PGREFTOP, see "NTPT_PGREFTOP" in **Optional Page-Specific Variables.**

Can I add information to an individual page's tag?

Yes. In addition to using the default fields that a page tag may contain, you can use the NTPT_PGEXTRA variable to add other fields to a specific page's tag. This variable is a tag modifier that is applied to the outgoing page tag. For example, if you want to include a color field set to "red" in the tag for a certain page, you would set NTPT_PGEXTRA to "color=red" in the HTML of the page you want to tag, before the page tag itself.

You can also modify existing fields in this way. For example, if you want to set the location (the lc field) of several pages to the same value, such as "welcome.html, " you can set the NTPT_PGEXTRA variable of each page to "lc=welcome.html." For details on NTPT_PGEXTRA, see "NTPT_PGEXTRA" in **Optional Page-Specific Variables**.

Can I add information to the tags for all the pages on my site?

Yes. You can use the NTPT_GLBLEXTRA variable to add information to the tags for all pages on your site. The NTPT_GLBLEXTRA variable works just like the NTPT_PGEXTRA variable, except that you will set it in the header of the ntpagetag.js file, and it will apply to all tagged pages. For details on NTPT_GLBLEXTRA, see "NTPT_GLBLEXTRA" in **Optional Variables in ntpagetag.js**.

When modifying an outgoing tag, the NTPT_GLBLEXTRA modifier is applied before the NTPT_PGEXTRA variable. Therefore, you can use the NTPT_PGEXTRA variable to override fields that are set in the NTPT_GLBLEXTRA variable.

Can I track Web applications (Special Services Tracking)?

The EMS can track Web applications using the POST method or the GET method, or a combination of both. If the Web application uses GET, no additional configuration of page tags is needed because parameters are encoded within the query string of the URL and will be automatically sent to the EMS and recorded. Applications using POST will require additional configuration because data are passed from the Web form/interface to the application transparently from the EMS perspective. To capture data from the POST method, the Data Provider's Web application must write parameters and values to a custom page tag JavaScript variable called NTPT_PGEXTRA. Once this variable has been populated, its contents will be passed to the EMS in the same manner as other page tag data (that is, as a query string appended to the URL). The application must populate NTPT_PGEXTRA on the resulting Web page created using the POST method. For example, if the Web application is a subsetter, a user may specify subsetting criteria such as GeographicExtent, SpectralBands, DateRange, DataType, etc., on a form input page and then click on a submit button. On the following confirmation page, the page tag variable NTPT_PGEXTRA should have information of value. See **Figure 3: Parameterized NTPT_PGEXTRA Example** below. An alternately parameterized page tag can also be specified for unsuccessful submissions.

The figure below has six parameter value pairs that will be captured by the EMS using the NTPT_PGEXTRA variable in the page tag JavaScript. The total number of characters that can be placed in the NTPT_PGEXTRA variable is limited to 1,500 including parameter names, parameter values, special symbols and delimiters.

```
NTPT_PGEXTRA='GeoExtentLat=23.5&GeocExtentLng=96.3&SpectralBands=1,2,3,4&D
ateRange=12/01/2005-12/31/2005&DataType=MOD12 &OrigToSubsetVolRatio=.10'
```

Figure 3: Parameterized NTPT_PGEXTRA Example

Figure 4 below provides an example of the JavaScript to be inserted into pages created using the POST method. Note that the only difference between the generic page tag and this example is the var NTPT_PGEXTRA line. Complete page tag installation and configuration instructions are located at http://ems.eos.nasa.gov/examples/ntpagetag.zip.

```
<!-- BEGIN: EMS Page Tag Notes -->
<!-- Some content Copyrighted 2004 Sane Solutions, LLC.  All
rights reserved. -->
<!-- Modified 02/07/2005 kjm -->
<!-- Page tag script to transfer data from POSTs-->
<!-- This information can be placed anywhere within your page
before the closing BODY tag -->
<!-- YOU MUST CHANGE src="/scripts/ntpagetag.js" to comply with
your local implementation -->
<!--Uncomment the 'var NTPT_PGEXTRA=""' line and insert
application specific parameter value pairs from POSTs -->
<!-- END: EMS Page Tag Notes -->
<!-- BEGIN: EMS Page Tag -->
<script language="JavaScript">
```

//Format Example

var NTPT_PGEXTRA='GeoExtentLat=23.5&GeocExtentLng=-96.3& SpectralBands=1-4&DateRange=12/01/2005-12/31/2005&DataType=MOD12&OrigToSubsetVolRatio=.10';

```
// var NTPT_PGREFTOP = false;
// var NTPT_NOINITIALTAG = false;
</script>
<script language="JavaScript"
src="/scripts/ntpagetag.js"></script>
<!-- END: EMS Page Tag -->

<!-- BEGIN: EMS Page Tag No Script -->
<noscript>
<img
src="http://ws1.ems.eosdis.nasa.gov/images/ntpagetag.gif?js=0"
height="1" width="1" border="0" hspace="0" vspace="0" alt="">
</noscript>
<!-- END: EMS Page Tag No Script -->
```

Error! Not a valid bookmark self-reference.,

8.3 Event Tagging

What is "event tagging"?

In addition to tagging the page when it loads, sometimes it is useful to generate a page tag when certain on-page actions occur. This type of page tagging is known as "event tagging."

Is event tagging different from page tagging?

Event tagging is a special type of page tagging. Generating an event tag on a page requires a different method of tagging.

What is event tagging used for?

You can use event tagging to analyze on-page events such as adding a product to a shopping cart, changing a field on a form, selecting an option in a drop-down list box, submitting a form and user selection of specific links. You implement event tagging by calling a JavaScript event tagging function when events you want to track occur.

At the most basic level, you can use event tagging to tag any JavaScript event that occurs on a page.

The ntpagetag.js file provides JavaScript functions for different types of event tagging.

What JavaScript functions does ntpagetag.js provide to facilitate event tagging?

To facilitate event tagging, use these functions:

To tag	**Use this function**
Link events	ntptLinkTag (for details, see “ntptLinkTag”)
Form submissions	ntptSubmitTag (for details, see “ntptSubmitTag”)
Generic events	ntptEventTag (for details, see “ntptEventTag”)

You can use the ntptAddPair and ntptDropPair functions to modify the outgoing tag before it is sent. For details on ntptAddPair, see “ntptAddPair” in **User-Callable Page Tagging Functions (Event Tagging)**. For details on ntptDropPair, see “ntptDropPair” in **User-Callable Page Tagging Functions (Event Tagging).**

How do I tag a link to a PDF, HDF, GIF, or JPEG file, or any other non-HTML document available for download on my site?

You cannot use standard page tagging to tag non-HTML files on your Web server because there is no way to include the JavaScript necessary to generate the tag within the file. However, you can tag a link to a file and have it appear in the page tag log in a manner similar to tagged pages.

The page that links to the non-HTML file must include the EMS NetInsight page tag, as usual. To tag the link to the non-HTML content, use the “onclick” attribute of the link anchor element to call the ntptLinkTag function. For example:

```
<a href="some.pdf" onclick="return ntptLinkTag( this );">Some
PDF</a>
```

For details on ntptLinkTag, see “ntptLinkTag” in **User-Callable Page Tagging Functions (Event Tagging)**.

How do I tag the submission of a form?

To tag the submission of a form, use the onsubmit attribute of the form element to call the ntptSubmitTag function. For example:

```
<form name="myform" method="post" onsubmit="return
ntptSubmitTag(this );">
```

For details on ntptSubmitTag, see “ntptSubmitTag” in **User-Callable Page Tagging Functions (Event Tagging)**.

Can I tag a partially completed form?

You can get information from a partially completed form into a tag by using the onchange method of the form input element to generate an event tag each time a form input element changes. For example:

```
<input name="mytext" type="text"
onchange="ntptEventTag('mytext_changedto=' + escape( this.value )
);"></input>
```

For details on ntptEventTag, see “ntptEventTag” in **User-Callable Page Tagging Functions (Event Tagging)**.

How can I tag other user events that happen on my page?

You can use the ntptEventTag function to tag arbitrary JavaScript events on a page. For example, you can use the “onclick” attribute of the img element if you want to generate a tag each time a user clicks a non-link image. For example:

```
<img src="/img/not-a-link.png" onclick="ntptEventTag(
'imgclick='+ escape( this.src ) );">
```

For details on ntptEventTag, see “ntptEventTag” in **User-Callable Page Tagging Functions (Event Tagging)**.

Why is there sometimes a delay when I click a tagged link or form submission on my page?

The events that the ntptLinkTag and ntptSubmitTag functions are tagging—following links and submitting forms—ultimately lead to the unloading of the existing page. The ntpagetag.js JavaScript file introduces a small delay when tagging these events to ensure that the tag is generated correctly before the page is unloaded.

You can configure this delay by setting the NTPT_MAXTAGWAIT variable in the header of the ntpagetag.js file. For details, see “NTPT_MAXTAGWAIT” in **Required Variables in ntpagetag.js**.

What if I only want to tag events on a page but not tag the page itself?

Sometimes you may only want to tag the events on a page but not tag the initial loading of the page. You can do this by setting the NTPT_NOINITIALTAG variable to “true.” The initial page tag will be suppressed but subsequent event tags on the page will fire. For details on NTPT_NOINITIALTAG, see “NTPT_NOINITIALTAG” in **Optional Page-Specific Variables**.

How can I tag events in a Flash presentation?

You can generate a page tag for any kind of end user action within the Flash presentation. For example, you could send a page tag every time the user performs any of the following actions:

- Releases a button
- Enters a key frame

- Rolls over a symbol
- Drags a symbol
- Presses but doesn't release a button
- Submits a Flash form
- Types a specific character on the keyboard
- Loads an extended URL
- Moves the mouse a specific distance in a specific direction

Before you can tag events in a Flash presentation, you must first follow the instructions provided in this document to include the page tagging script in the HTML page that contains the Flash (.swf) document. (See "**EMS NetInsight Page Tagging Script**" in Chapter 2.)

Thereafter, the ntptEventTag function can be called at the point in the presentation where you want a page tag sent. For more information, see "ntptEventTag" in **Required Variables in ntpagetag.js.**

For example, to send a page tag every time a user releases a particular button, enter the following in the Actions Inspector (on the Window menu, click Actions) for the button:

```
on (release) {
loadMovieNum("javascript:ntptEventTag('button=released')",1);
}
```

Note: When you call ntptEventTag, specify a movie layer whose number is larger than the largest layer you are using for your Flash presentation. If you do not specify a layer, you will replace the page that contains the Flash presentation (layer 0) with a page containing only the result of the JavaScript (which would read "true").

How Does EMS NetInsight Treat Event Tags?

EMS NetInsight treats event tags the same way it treats normal page tags. If you want to report on the fields included in an event tag, you must define a custom Page Tag parameter (contact ems-help@lists.nasa.gov for help).

Note: By default, an event tag will be counted as a page view unless you specify a value of "0" for the "pv" field (that is, "pv=0").

8.4 Importing Data into EMS NetInsight

Where is the information gathered by page tagging stored?

When you have correctly inserted the EMS NetInsight page tagging script into the HTML source of each page on the site you want to tag and have copied the ntpagetag.js file to the proper location, a tag will be generated each time a user requests a tagged page on your site. This tag is stored in the log file for the Web server that is serving the ntpagetag.gif image (for example, http://ws1.ems.eosdis.nasa.gov).

How do I know whether my page tags are collecting information?

Please contact the EMS Team at ems-help@lists.nasa.gov to request a check of the local log files to determine if your data is being collected.

How do I configure an EMS NetInsight profile for use with page tagging logs?

Please contact the EMS Team at ems-help@lists.nasa.gov. Only the EMS Team has the required permissions to set up new EMS NetInsight profiles.

How does EMS NetInsight report on the data collected from page tags?

EMS NetInsightuses the location (“lc”) and referrer (“rf”) fields parsed from the page tag as the page and referrer for all requests. The Screen Resolution Summary reports on users’ screen resolutions based on the value parsed from the page tag’s “rs” field.

EMS NetInsight uses the value of the “sc” field as the status code of the request (for example, a value of “200” indicates a successful request). Note that if a page tag contains the “sc” field and its value is greater than or equal to 400 (such as “sc=404”), then the request will appear with that status code on the EMS NetInsightError Summary.

If the “pv” field is set to “pv=1,” a page tag is counted as a page view in EMS NetInsight—even when EMS NetInsight is configured to augment only log files with additional data, which normally discards the page tag request when tracking page views. For the other page tagging options, a page tag is counted as a page view in EMS NetInsightby default. Conversely, if the “pv” field is set to “0,” a request for the page tag is not counted as a page view in EMS NetInsight if it otherwise would be counted.

You can use EMS NetInsight to report on additional data by defining a Page Tag parameter. For details, refer to “Tracking Parameters in Page Tags” in the EMS NetInsight® User’s Guide. Please note that the EMS is required to limit proprietary NetInsight documentation to EMS customers. Please contact ems-help@lists.nasa.gov to request authorization to access these documents.

8.5 Common Report Generation Problems

Why am I unable to generate a custom report for my Web application parameter(s)?

The inability to generate a custom report is most frequently caused by the following issues:

- The parameter was not included in the page tag manifest.
- A new parameter was added, but the updated page tag manifest was not sent to EMS staff.
- The Web applications host was not included in the page tag manifest.
- If the Web application was run from a number of different hosts, then either the complete list of these hosts was not provided to EMS staff or the parameter name changed depending on which server was hosting the Web application.
- The Web application generated a file that was not tagged, such as text output or an image.

Why am I unable to create a report showing what FTP distributed products are ordered using my tagged Web page interfaces/clients?

The EMS is not able to make the association when an HTTP Web order application hands off a requested product order to an FTP distribution system. Data Provider FTP distribution logs are sent to the EMS using a method different from Web page tag collection.

My products are distributed using a tagged HTTP client interface, but I'm still not able to generate reports on which clients are used to order my products.

If the client interface is generated using the POST method, make sure the Web application's parameters and values are included in a custom page tag JavaScript variable called "NTPT_PGEXTRA."

Common oversights, such as those listed in the answer to the first question in this section, may be impacting report results. Please contact the EMS staff at ems-help@lists.nasa.gov who will work with you to identify and resolve problems.

I have been asked to change my product ordering Web applications so management can generate reports on what products are ordered using different clients. There are numerous applications on many different servers. Is there an easy way to do this?

Many organizations employ single files with information common to all their Web pages, such as a header, footer or contact information. Use one of these files to include the NetInsight page tag. This code snippet can be found in the file, sample.html (part of the ntpagetag.zip file, described in **EMS Page Tagging Components** in this document.

8.6 Other Issues

Why have I suddenly started getting ssh connection errors when transferring my log files to EMS when I haven't changed anything?

The EOS Backbone Network (EBNet) security procedures require registration of the IP addresses of all data provider hosts transferring files to the EMS. This is the most likely cause of error messages such as the following:

```
ssh: connect to host ws1.ems.eosdis.nasa.gov port 22: Connection
timed out
```

This is the result of a change in your host's IP address, such as a move to a new server, without notifying the EMS of the change.

This page is intentionally left blank.

Appendix A: Page Tag Manifest

In order for the EMS to capture and report on information collected from page tags, Data Providers need to complete the Page Tag Manifest (an Excel file). The manifest has two sections. In the first section, the Data Providers identify the domain names which have been tagged with the EMS page tags. The second section is used to capture information in query strings embedded in the URL from Web applications using POST and GET methods. An embedded URL is shown below:

```
http://<domain name>/<Program Name>?<Parameter1>=<value1>&<Parameter
2>=<value2>&<ParameterN>=<valueN>
```

For each parameter, the Data Provider must provide the name and a brief description in the Page Tag Manifest file. The table below describes the format of the file. A template of the Page Tag Manifest file is available from http://ems.eos.nasa.gov/templates/PageTagManifestTemplate.xls.

EMS Page Tag Manifest File

Page Tag Manifest *Please replace the example entries below with information specific to your environment. **Domain Name Identification Section**	
INSTRUCTIONS: -Use this section to identify all domains where page tags are installed. -Metrics will only be available for domain names listed below. -Please include development and testing domains. They can be removed later. -Insert as many rows as necessary.	
Domain Name	**Description**
www.foo.com	Main Site
mirror.foo.com	Main Site mirror
dev.foo.net	Development site

Web Application and Parameter Identification Section			
Instructions: -Use this section to identify all Web application URLs that should be tracked by the EMS. -The fully qualified URL is needed including the Web application name. -For each parameter in the request (POST or GET) fill in the corresponding parameter section and provide a short description. -Specify the method (POST or GET). -Parameter entries are case sensitive and must be entered exactly as they appear in the URL. -If parameters are not included on this manifest, they will not be available for reporting. -Insert as many rows as necessary.			
Domain and Program Name (URL)	http://www.foo.com/cgi-bin/myApp.pl		
Description	Dynamically creates plots of optical thickness from a variety of atmospheric retrievals		
	Parameter	**Method**	**Description**
Parameters 1-3	Param 1	POST	…
	Param 2	POST	…
	Param 3	POST	…
Domain and Program Name (URL)	http://www.foobar.com/cgi-bin/anotherApp.pl		
Description	Orders products by date, time, mission, product and version		
Parameters 1-5	Param 1	GET	…
	Param 2	GET	…
	Param 3	GET	…
	Param 4	GET	…
	Param 5	GET	…

Figure 4: Page Tag Manifest Example

Figure 6 on the next page presents an example of a Page Tag Manifest file completed with sample data from the Physical Oceanography DAAC (PO.DAAC). The “Web Application and Parameter Identification Section” was created based on the GET request below. POSTs will be handled in the same manner.

Example of GET Request – PO.DAAC: *http://poet.jpl.nasa.gov/cgi-bin/esip/data_access_frame.pl?&BBOX=-100.0,10.0,-70.0,35.0&TIME=2004-12-31/2004-12-1&LAYERS=sst&INSTRUMENT=pathfinder5&REQUEST=GetMap&HOUR=0&QUALITY=4&ALG=0&FORMAT=gif&WIDTH=640&HEIGHT=480&MINVALUE=2.0&MAXVALUE=32.0&PLOTTITLE=Sea+Surface+Temperature+%28C%29&PAGETITLE=*

Figure 5: PO.DAAC Get Method Example

PODAAC Example Page Tag Manifest		
*Please replace the example entries below with information specific to your environment.		
Domain Name Identification Section		
INSTRUCTIONS: -Use this section to identify all domains where page tags are installed. -Metrics will only be available for domain names listed below. -Please include development and testing domains. They can be removed later. -Insert as many rows as necessary.		
Domain Name	**Description**	
http://podaac.jpl.nasa.gov	Main Site	
http://poet.jpl.nasa.gov	The PO.DAAC Ocean ESIP Tool (POET)	
http://nereods.jpl.nasa.gov	Near-real-time image distribution server	
Web Application and Parameter Identification Section		
Instructions: -Use this section to identify all Web application URLs that should be tracked by the EMS. -The fully qualified URL is needed including the Web application name. -For each parameter in the request (POST or GET) fill in the corresponding parameter section and provide a short description. -Specify the method (POST or GET). -Parameter entries are case sensitive and must be entered exactly as they appear in the URL. -If parameters are not included on this manifest, they will not be available for reporting. -Insert as many rows as necessary.		
Domain and Program Name (URL)	http://www.foo.com/cgi-bin/myApp.pl	
Description	Dynamically creates plots of optical thickness from a variety of atmospheric retrievals	

	Parameter	Method	Description
Parameters 1-13	BBOX	GET	Bounding box in DD
	TIME	GET	Search date range
	LAYERS	GET	Layers used in visualization
	INSTRUMENT	GET	Name of instrument
	REQUEST	GET	Type of action preformed
	HOUR	GET	Time of day
	FORMAT	GET	Output format (GIF, GZ, etc)
	WIDTH	GET	Output size width
	HEIGHT	GET	Output size height
	MINVALUE	GET	Minimum plot value
	MAXVALUE	GET	Minimum plot value
	PLOTTITLE	GET	Title of plot
	PAGETITLE	GET	Title of page

Figure 6: Page Tag Manifest File Example Using PODAAC Information

Page Tag Manifest File Naming Convention

All Data Provider Page Tag Manifest Files must use the following convention:

`<YYYYMMDD>_ <Provider>_ PageTagManifest.xls`

where

YYYY	designates the 4 digit year for the time the Page Tag Manifest File was created
MM	designates the 2 digit month, 01 through 12
DD_	designates the 2 digit day, 01 through 31, followed by an underscore

Provider_ designates the provider of the data, a mutually agreed upon acronym defined in the Operations Agreements between the EMS and the Data Providers, followed by an underscore.

PageTagManifest.xls indicates that the file is the Page Tag Manifest File worksheet

Page Tags Updates

Data Providers are expected to change Web servers, Web pages, applications and configurations over time. The only requirement Data Providers must adhere to during upgrades and migrations is to ensure that page tags are carried over from one system to the next, or are added to new systems. Because of the generic nature of basic page tag configurations, the only element that needs to be changed for the implementation will be references to the JavaScript source to reflect the new operating environment. If a Web application or event page tags are used, they may have to be modified for the new environment per instructions in previous sections and associated documentation.

The EMS must be notified in the event of changes in configuration of domains serving page tags or changes to parameters captured by page tags. Each time a change in configuration occurs, Data Providers should notify the EMS Team two weeks prior to migration so accounts can be modified and metrics can continue to be captured. To facilitate this process, Data Providers must provide the EMS with two manifests. The first file, the Page Tag Manifest, will contain the new configuration including carryovers from the previous manifest. The second file, the Invalid Page Tag Manifest, will contain domains and parameters that should no longer be captured by the EMS. A template of this document is available on the EMS Web site at http://ems.eos.nasa.gov/templates/PageTagManifestTemplate.xls.

The name of the Invalid Page Tag Manifest file must be in the following format for all Data Providers:

```
<YYYYMMDD>_< Provider>_ InvalidPageTagManifest.xls
```

where

YYYY	designates the 4 digit year for the time the Invalid Page Tag Manifest File was created
MM	designates the 2 digit month, 01 through 12
DD_	designates the 2 digit day, 01 through 31, followed by an underscore
Provider_	designates the provider of the data, a mutually agreed upon acronym defined in the Operations Agreements between the EMS and the Data Providers, followed by an underscore.
InvalidPageTagManifest.xls	indicates that the file is the Invalid Page Tag Manifest worksheet